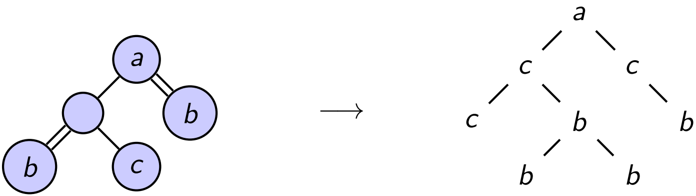
Consistency of injective tree patterns

Claire David   Nadime Francis   Filip Murlak

U. Paris-Est Marne   ENS Cachan   U. Warsaw

FIT 2015, Warsaw

Satisfiability of tree patterns: P or NP?



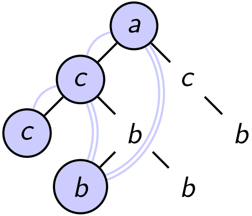
Problem

Can given pattern π be matched in some tree from regular language L?

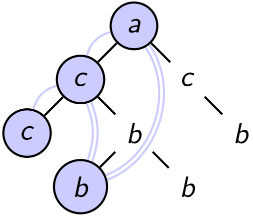
Semantics 1: homomorphisms

Pattern π is matched in tree T if there is a homomorphism $h\colon \pi \to T$.

(Preserves labels, child and descendant relations.)



Semantics 1: homomorphisms



Pattern π is matched in tree T if there is a homomorphism $h\colon \pi \to T$.

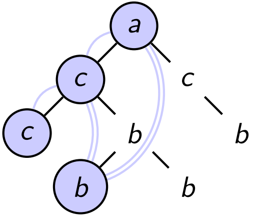
(Preserves labels, child and descendant relations.)

- XPath(/,//, []), query language for XML docs [Miklau, Suciu '02]

```
a[//b]/[//b,c]
```

(XML docs are trees)

Semantics 1: homomorphisms



Pattern π is matched in tree T if there is a homomorphism $h\colon \pi \to T$.

(Preserves labels, child and descendant relations.)

- XPath(/,//, []), query language for XML docs [Miklau, Suciu '02]

  `a[//b]/[//b,c]`

  (XML docs are trees)

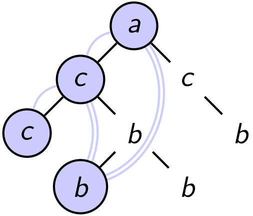
- Conjunctive queries over trees [Gottlob, Koch, Schultz '04]

$$\exists x_1 \cdots \exists x_5\ a(x_1) \wedge b(x_3) \wedge b(x_4) \wedge c(x_5) \wedge \\ \wedge\ child(x_1, x_2) \wedge desc(x_1, x_3) \wedge \\ \wedge\ desc(x_2, x_4) \wedge child(x_2, x_5)$$

Semantics 1: homomorphisms

Pattern π is matched in tree T if there is a homomorphism $h\colon \pi \to T$.

(Preserves labels, child and descendant relations.)



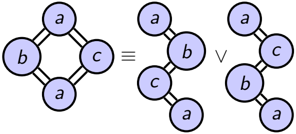
- XPath(/,//, []), query language for XML docs [Miklau, Suciu '02]

  `a[//b]/[//b,c]`

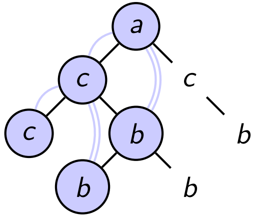
  (XML docs are trees)

- Conjunctive queries over trees [Gottlob, Koch, Schultz '04]

$$\exists x_1 \cdots \exists x_5\ a(x_1) \wedge b(x_3) \wedge b(x_4) \wedge c(x_5) \wedge \\ \wedge\ \mathit{child}(x_1, x_2) \wedge \mathit{desc}(x_1, x_3) \wedge \\ \wedge\ \mathit{desc}(x_2, x_4) \wedge \mathit{child}(x_2, x_5)$$



Semantics 2: injective



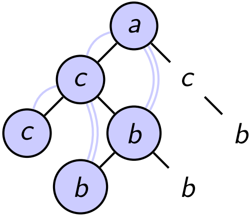
Pattern π is matched in tree T if there is an injective homomorphism $h \colon \pi \to T$.

(All nodes mapped to different tree nodes.)

Semantics 2: injective

Pattern π is matched in tree T if there is an injective homomorphism $h\colon \pi \to T$.

(All nodes mapped to different tree nodes.)



- incomplete information about XML docs in DOM model
  [Barceló, Libkin, Poggi, Sirangelo '09]
  - nodes have unique IDs
  - labels and relations may be lost
  - when ID is lost, node is lost

  pattern π: incomplete XML doc;
  regular language L: correct docs (schema);
  π is satisfiable iff incomplete doc extends to a correct doc

Regular languages

Sets of finite labelled unranked trees

- definable in monadic second order logic;
- recognizable by automata;
- generated by RelaxNG (covers DTD, XSM).

Regular languages

Sets of finite labelled unranked trees

- definable in monadic second order logic;
- recognizable by automata;
- generated by RelaxNG (covers DTD, XSM).

Define several types of trees, each specified (recursively) by

- the label of the root,
- possible sequences of immediate subtree types (regexp);

and choose some of the types to form your language.

Regular languages

Sets of finite labelled unranked trees

- definable in monadic second order logic;
- recognizable by automata;
- generated by RelaxNG (covers DTD, XSM).

Define several types of trees, each specified (recursively) by

- the label of the root,
- possible sequences of immediate subtree types (regexp);

and choose some of the types to form your language.

Example: *a*-only path from root to leaf, *b*'s elsewhere

- type τ: root label *a*, immediate subtree types $\sigma^*\tau\sigma^* + \epsilon$;
- type σ: root label *b*, immediate subtree types σ^*;
- choose: τ

Satisfiability of tree patterns: complexity landscape

Semantics	Consistency (fixed L)	Satisfiability (given L)
Homomorphism		
Injective		

Satisfiability of tree patterns: complexity landscape

Semantics	Consistency (fixed L)	Satisfiability (given L)
Homomorphism		NP-complete Can find polynomial witness. Can reduce NP-complete problem: does given regular expression α generate a word using all letters in α.
Injective		

Satisfiability of tree patterns: complexity landscape

Semantics	Consistency (fixed L)	Satisfiability (given L)
Homomorphism	NP-complete, even child-only [Benedikt, Fan, Geerts '05]	NP-complete Can find polynomial witness. Can reduce NP-complete problem: does given regular expression α generate a word using all letters in α.
Injective		

Satisfiability of tree patterns: complexity landscape

Semantics	Consistency (fixed L)	Satisfiability (given L)
Homomorphism	NP-complete, even child-only [Benedikt, Fan, Geerts '05]	NP-complete Can find polynomial witness. Can reduce NP-complete problem: does given regular expression α generate a word using all letters in α.
Injective	?	

Zooming in on consistency under injective semantics

[Barceló, Libkin, Poggi, Sirangelo '09]

- PTIME without descendant.
  - Can given set of *words* be matched disjointly in a *word* from L?

Zooming in on consistency under injective semantics

[Barceló, Libkin, Poggi, Sirangelo '09]

- PTIME without descendant.
  - Can given set of *words* be matched disjointly in a *word* from L?

[Kopczyński '11]

- PTIME with ≤ 1 descendant on each branch.
  - Can given set of *trees* be matched disjointly in a *tree* from L?

Zooming in on consistency under injective semantics

[Barceló, Libkin, Poggi, Sirangelo '09]

- PTIME without descendant.
  - Can given set of *words* be matched disjointly in a *word* from L?

[Kopczyński '11]

- PTIME with ≤ 1 descendant on each branch.
  - Can given set of *trees* be matched disjointly in a *tree* from L?

[David, Francis, Murlak '14] (this work)

- ≤ 2 descendants on each branch: NP-complete;

Zooming in on consistency under injective semantics

[Barceló, Libkin, Poggi, Sirangelo '09]

- PTIME without descendant.
  - Can given set of *words* be matched disjointly in a *word* from L?

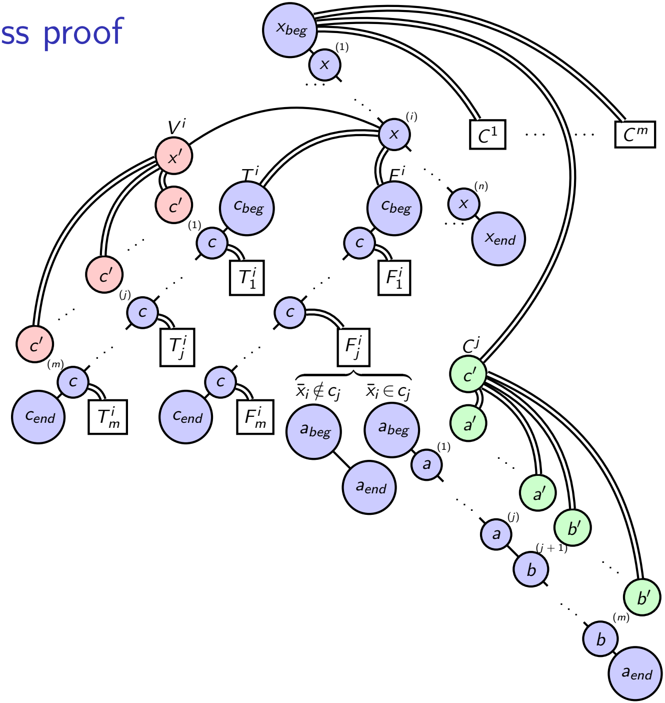
[Kopczyński '11]

- PTIME with ≤ 1 descendant on each branch.
  - Can given set of *trees* be matched disjointly in a *tree* from L?

[David, Francis, Murlak '14] (this work)

- ≤ 2 descendants on each branch: NP-complete;
- PTIME (FPT) with descendant only (without child)
  - works for injective and homomorphism semantics,
  - extends to patterns with sibling order (following-sibling).

On the hardness proof



Challenges

1. Our FPT algorithm has complexity $2^{2^{\|L\|}} \cdot \mathrm{poly}(\|\pi\|)$; the NP upper bound gives complexity $2^{\mathrm{poly}(\|L\|,\|\pi\|)}$. Inherent trade-off, or can this be reconciled?

2. Assuming we stick to descendant and following sibling, is consistency still tractable for patterns that are DAGs, not trees?